Le Web Embedding, la vraie révolution cachée derrière RankBrain

Par Philippe Yonnet

Domaine :	Recherche	**Référencement**
Niveau :	Pour tous	**Avancé**

L'annonce par Google de l'intégration dans son algorithme de classement de "Rankbrain", une brique logicielle qui selon ses créateurs embarque de l'intelligence artificielle, a fait couler beaucoup d'encre dans le petit monde du SEO. Beaucoup de commentateurs ont spéculé sur ce que pouvait entrainer l'emploi de l'intelligence artificielle dans un algorithme de moteur de recherche. Mais en réalité, la véritable révolution apportée par Rankbrain, c'est l'exploitation d'une méthode nouvelle et très prometteuse en linguistique informatique : le "word embedding". Et dans ces méthodes, le rôle de l'intelligence artificielle est assez mineur : les outils d'IA sont simplement utilisés pour "percevoir" des valeurs concernant des termes, des phrases ou des documents. Mais cela nous fait parfaitement comprendre la façon dont Google fonctionne aujourd'hui et surtout la direction qu'il prendra à l'avenir. Et donc les méthdes SEO à mettre en œuvre pour prendre en compte cette nouvelle vision. Décryptage...

Pour comprendre ce qu'est le "Word Embedding", et pourquoi ce concept est révolutionnaire, nous vous proposons de revenir aux concepts fondateurs qui ont été employés depuis des dizaines d'années par les moteurs de recherche. Et nous verrons ensuite en quoi cette nouvelle approche représente un "bond" technologique" majeur qui annonce une adoption rapide, et des applications dans de nombreux domaines nouveaux.

Un peu d'histoire sur les méthodes utilisées en linguistique informatique

L'analyse en sac de mots (bag of words)

Les méthodes utilisées encore aujourd'hui par les principaux moteurs de recherche trouvent leur origine dans des travaux menés dans les années 50. Ce paradigme aura donc tenu plus d'un demi-siècle sans être réellement bousculé dans ses fondements, essentiellement parce que ses applications étaient les seules qui pendant toute cette période permettaient des applications viables.

Sur le plan théorique, l'approche s'appuie sur les travaux du linguiste Noam Chomsky. Zelig Harris, un autre linguiste, a posé les principes de la méthode en 1954. En

pratique, l'idée consiste à faire une analyse statistique de la fréquence d'apparition (la fréquence d'occurrence) des termes au sein des textes.

Pour faire ce travail, les occurrences d'un même terme sont comptées sans tenir compte de l'ordre d'apparition des termes dans le texte, ni de la phrase qui le contient : l'information n'est pas conservée. C'est pour cela que les pionniers de cette méthode l'ont appelé « bag of words » : le « sac de mots ».

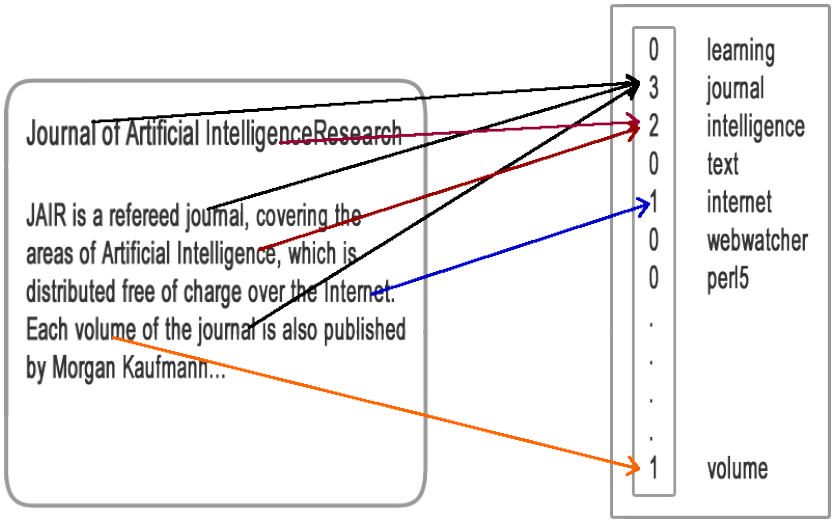

Fig.1. Principe de l'analyse en sac de mots :
les termes sont comptés, puis les statistiques d'occurrences analysées.

*Le modèle vectoriel, tf*idf, et le Cosinus de Salton*

Au début des années 70, Gérard Salton introduit l'idée d'utiliser un « poids » pour les termes qui tient compte de la fréquence d'occurrence d'un terme dans tous les documents. Ce poids, qu'il a baptisé tf*idf, fournit une mesure de l'importance relative d'un terme dans un document. A la base, les calculs des poids reposent sur des statistiques (fréquences d'occurrences) calculés à partir d'une analyse en « sac de mots ». Nous en avons déjà parlé dans cette lettre.

Gérard Salton introduit également l'idée de représenter les documents sous formes de vecteurs mathématiques (modèle vectoriel). Les coordonnées de ces vecteurs sont fournies par les poids tf*idf. Cette idée s'est avérée particulièrement fructueuse en linguistique informatique, car les ordinateurs sont très efficaces pour réaliser des calculs sur des vecteurs ou des matrices. Or, cette représentation vectorielle des documents fournit un moyen très simple d'identifier dont le contenu est « proche » : il suffit de mesurer leur distance angulaire, ce qui revient à calculer le cosinus de l'angle entre les deux vecteurs représentant le document. Cette méthode (et ses variantes) baptisée « Cosinus de Salton », a été utilisée ensuite dans de nombreux moteurs de recherche. Là aussi, voir les articles parus à ce sujet précédemment dans cette lettre.

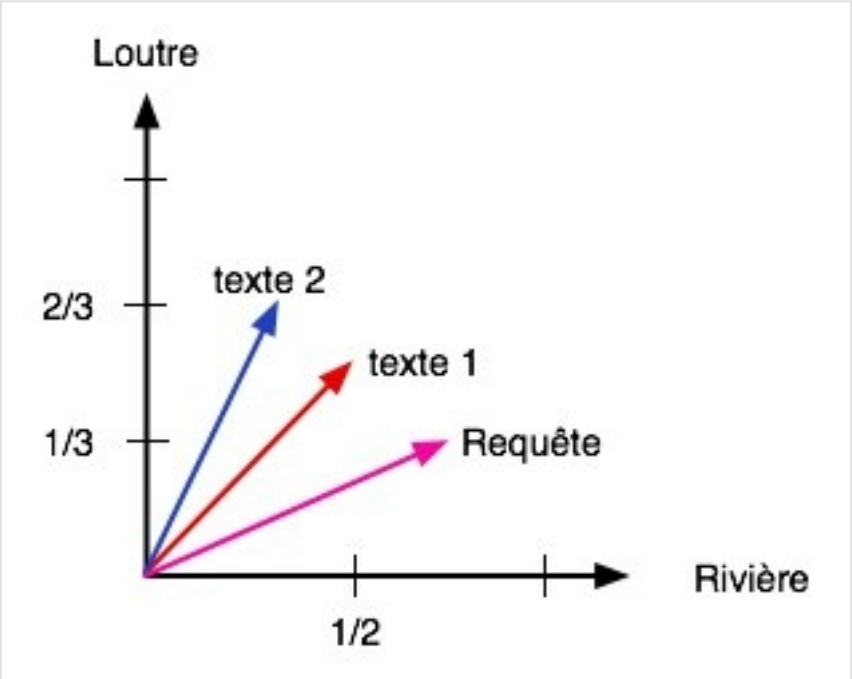

Fig.2. La méthode du cosinus de Salton illustrée : la requête est un document, on peut donc mesurer sa distance angulaire avec les documents figurant dans l'index du moteur. Les axes correspondent aux termes figurant dans l'index. Le modèle vectoriel travaille donc sur un espace qui a autant de dimensions que de termes, ce qui rend les calculs assez complexes, mais réalisables avec un ordinateur.
Source de l'image : https://freres.peyronnet.eu/modele-vectoriel-et-cosinus-de-salton/

La méthode du « Cosinus de Salton » a néanmoins un inconvénient majeur : elle est incapable de gérer l'existence de « synonymes ». Par conséquent, deux documents utilisant des termes synonymes mais ayant le même sens seront considérés comme éloignés par le modèle, alors qu'en fait, ils ont le même sens. Et on trouve beaucoup trop de documents qui apparaissent comme « orthogonaux », parce qu'ils ne contiennent pas les mêmes termes.

Cette limite du modèle vectoriel de Salton constitue un défaut majeur dans un algorithme de moteur de recherches, mais il nous a fallu vivre avec en utilisant en particulier Altavista dans les années 90, mais aussi Google jusqu'à une époque très récente. La solution consiste évidemment à essayer de mieux tenir compte du sens des termes. Mais comment le faire avec des statistiques ?

Les méthodes LSI et LDA

A la fin des années 80, des linguistes se sont intéressés à l'analyse des statistiques de cooccurrence (quelle est la fréquence d'apparition de deux termes, trois termes etc. dans les documents ?). Cela revient à faire une analyse en composantes principales (ACP) sur les données statistiques issues d'une analyse en sac de mots, ou plus généralement, une décomposition en valeurs singulières (SVD). Avec cette approche, on obtient des corrélations entre termes, corrélations qui sont indicatives (mais hélas pas toujours) d'une proximité sémantique entre ces termes.

A = U x S x Vt

	d1	d2	d3	d4
a	6	7	1	0
b	8	6	0	1
c	6	9	8	5
d	0	1	8	8
e	2	0	9	7
f	2	0	7	7

	f1	f2	f3	f4
a	0.24	-0.51	0.08	0.06
b	0.25	-0.54	-0.64	-0.23
c	0.58	-0.28	0.57	0.13
d	0.42	0.37	0.16	-0.68
e	0.44	0.34	-0.24	0.66
f	0.39	0.29	-0.40	-0.09

	f1	f2	f3	f4
f1	23.1	0	0	0
f2	0	14.3	0	0
f3	0	0	3.5	0
f4	0	0	0	1.5

	d1	d2	d3	d4
f1	0.37	0.38	0.65	0.53
f2	-0.55	-0.63	0.37	0.38
f3	-0.69	0.59	0.27	-0.21
f4	0.26	-0.29	0.59	-0.69

Fig.3. Un exemple de calcul de décomposition en valeurs singulières sur une matrice. Le problème est qu'il faut travailler sur des matrices de très grandes dimensions produites par le modèle vectoriel traditionnel !

Cette approche a donné lieu à l'invention de plusieurs techniques :
• LSA (*Latent Semantic Analysis*) / LSI (*Latent Semantic Indexing*) (Deerwester 1988) , puis pLSA (*probabilistic LSA*) ;
• LDA (*Latent Dirichlet Allocation*) (Blei et al. 2003) qui permet d'identifier à quel sujet se rapporte un terme.

Ces méthodes reposent hélas sur des calculs qui ne sont pas du tout « scalables ». La force de calcul nécessaire rapportée au manque de précision des résultats obtenus ont de facto limité l'utilisation de ces approches dans les moteurs de recherche.

La solution : la sémantique distributionnelle ?

Les linguistes ont depuis longtemps une intuition : il faut tenir compte des mots qui entourent un terme pour en détecter le sens. Dès les premiers concepts posés par Zélig Harris en 1954, on retrouve cette idée que mesurer la façon dont les autres termes sont distribués autour du terme étudié, permet de dégager des informations sur ce que ce terme représente (son sens).

(19.1) A bottle of *tesgüino* is on the table.
Everybody likes *tesgüino*.
Tesgüino makes you drunk.
We make *tesgüino* out of corn.

Fig.4. Un exemple devenu classique dans la littérature scientifique autour de la sémantique distributionnelle. Ces différentes phrases permettent de deviner que le « tesgüino » doit être une boisson alcoolisée !

Maintenant, comment faire pour analyser des textes de façon automatique, et « calculer » ce genre d'informations ? Pendant très longtemps, tous les chercheurs monde ont buté sur des difficultés pratiques, et cette approche est restée très théorique.

Mais le développement récent et extraordinaire des technologies de type « réseaux de neurones » a fourni une solution radicale, simple et étonnamment efficace à un problème que certains jugeaient insolubles. Les premiers travaux démontrant la faisabilité de la méthode datent de 2010, les premières implémentations concrètes sont apparues en 2013 (Word2vec de Google), et depuis deux ans, la communauté des chercheurs en linguistique informatique se passionnent pour cette approche, le nombre de « papiers scientifiques » sur le sujet ayant connu une véritable explosion l'année dernière.

L'avènement du Word Embedding

Le word embedding, c'est quoi ?

La solution est de demander à un « réseau de neurones » de calculer une représentation numérique d'un terme. Cette représentation numérique est une série de coordonnées, permettant d'obtenir, cette fois-ci encore un vecteur représentant le terme. Par contre, le nombre de dimensions nécessaires pour « placer » un terme dans cet espace virtuel est beaucoup plus réduit que dans le modèle vectoriel de Salton, ce qui crée un avantage considérable pour cette méthode.

Le réseau de neurones utilisé « apprend » les coordonnées en observant un corpus (un ensemble des documents). Plus ce corpus est vaste, plus les coordonnées seront précises. Ce qu'analyse le programme, ce sont les fameuses distributions de termes.

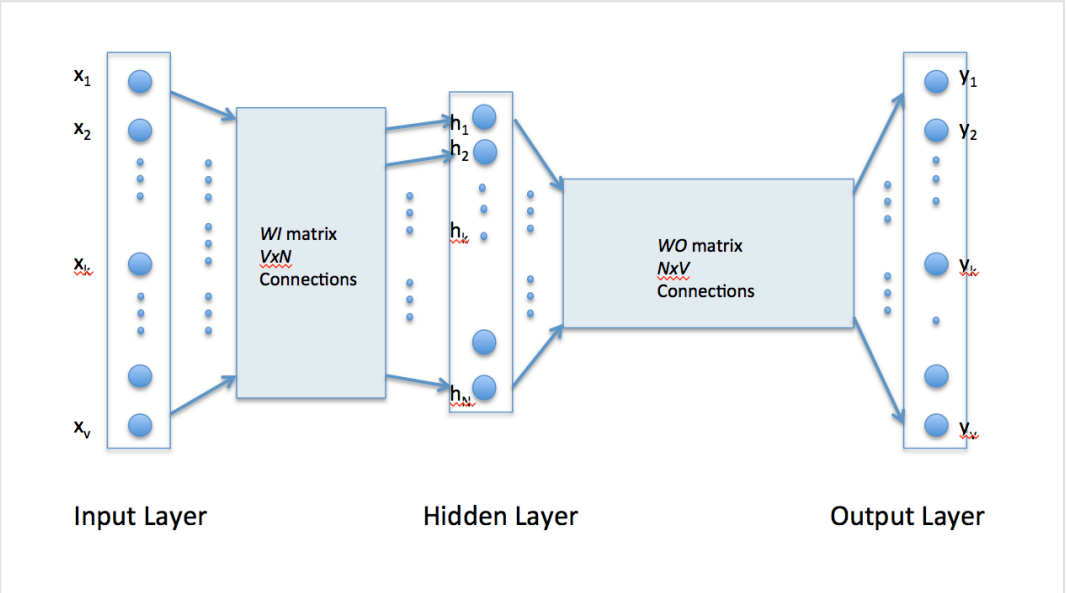

Fig.5. Un schéma représentant le réseau de neurones de Word2vec, l'outil inventé par Google pour le word embedding. Il s'agit d'un réseau de neurones dit « shallow », qui utilise peu de couches d'abstraction, et est donc peu complexe et peu gourmand en ressources quand on le fait tourner.

Une précision terminologique

Le terme « embedding » dans « word embedding » est une terminologie héritée du monde de l'intelligence artificielle, que l'on peut traduire par « incorporation de terme ». Les linguistes préfèrent « word representation », représentation de terme, mais il s'agit en fait de la même chose. Le « word embedding », c'est donc la représentation numérique d'un terme.

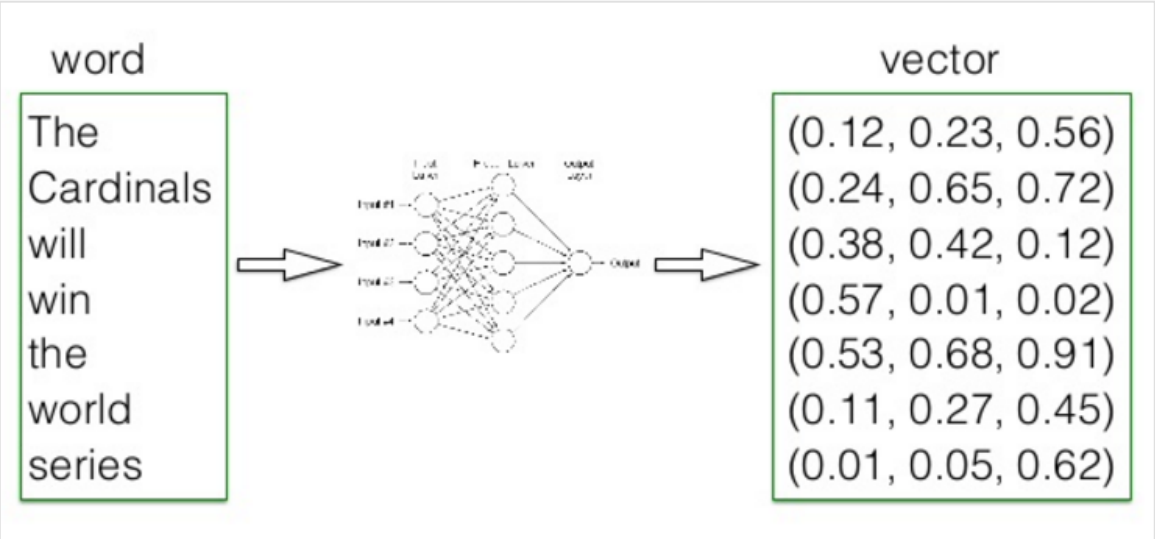

Fig.6. Illustration du concept de word embedding : après analyse du texte par un réseau de neurones, les termes à gauche sont représentés numériquement par un vecteur à 3 coordonnées !

Du « word » embedding au « sentence » embedding

Comme la représentation d'un terme est un vecteur, on peut procéder à des opérations arithmétiques sur ces vecteurs, pour créer un nouveau vecteur représentant :

- Un groupe de termes ;
- Ou un document.

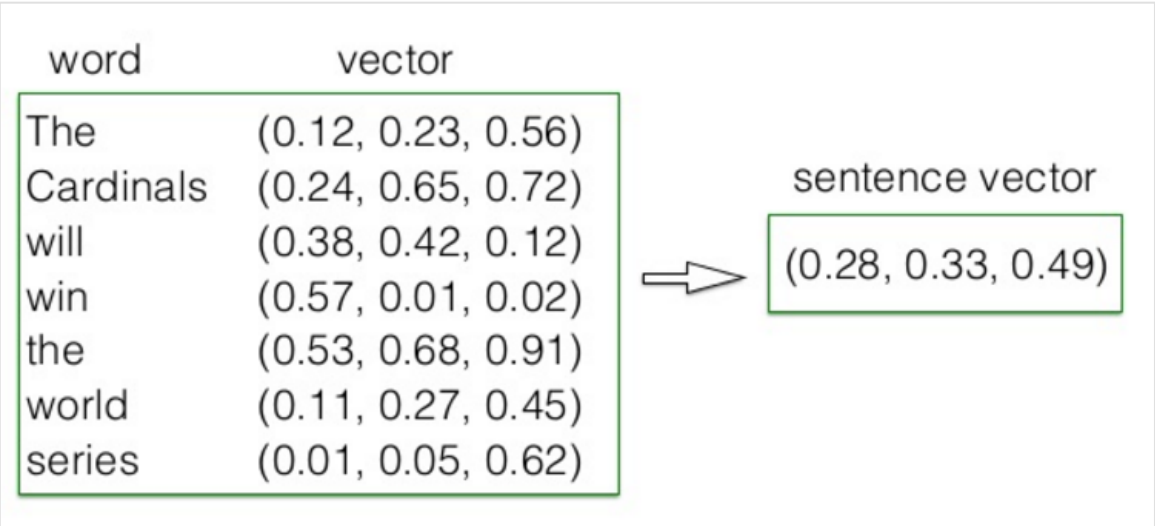

Fig.7. Représentation d'une phrase en partant de termes.

Et bien sûr, comme il s'agit d'un vecteur, on peut aussi calculer une « distance angulaire » entre termes, groupe de termes, et documents (en utilisant la méthode du Cosinus).

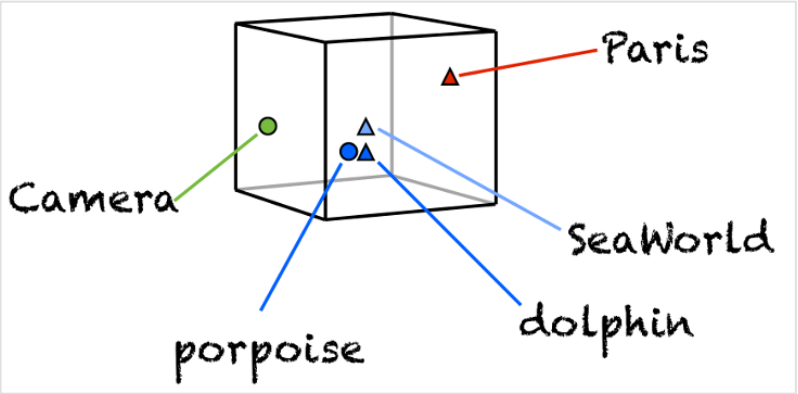

Fig.8. Exemple de word embeddings dans un espace à 3 dimensions : seaworld, dolphin, et porpoise apparaissent comme des termes très proches.

Les propriétés étonnantes des word embeddings

On peut alors procéder à des calculs arithmétiques simples sur des vecteurs : addition, soustraction, produit scalaire etc. Si on procède à des opérations arithmétiques entre word embeddings, on peut découvrir de nouveaux termes synonymes.

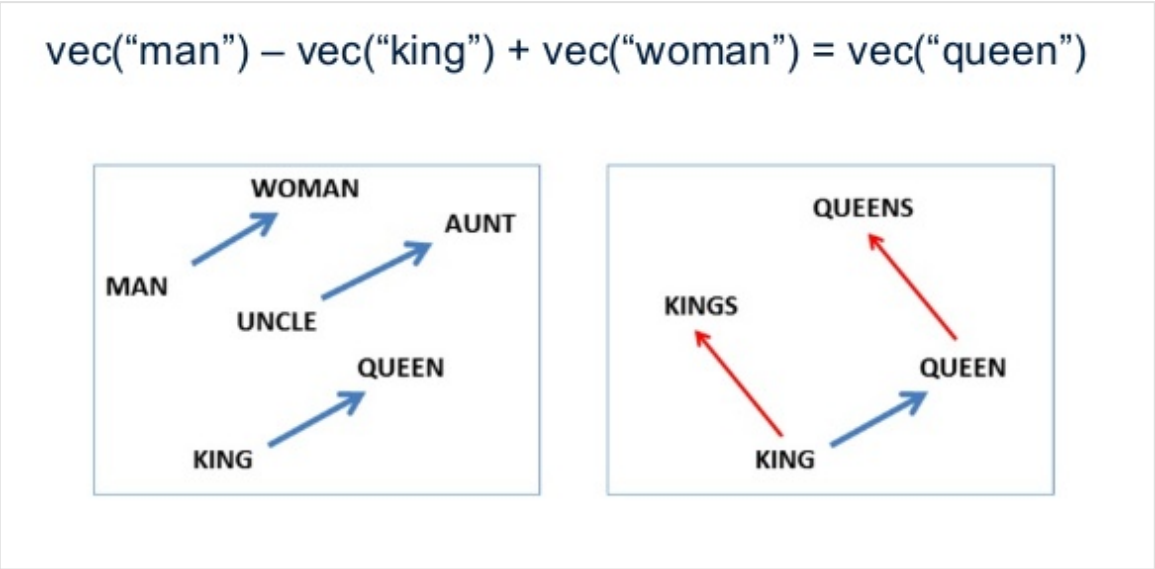

Fig.9. Exemples de calculs arithmétiques entre différents termes.

Quelles applications ?

Les applications des word embeddings sont nombreuses, et rappelons que nous n'en sommes qu'au début des travaux sur ce nouveau champ de recherche (dont les applications pratiques datent réellement de 2013).

Mais les champs d'application peuvent être regroupés en trois familles :
• Les applications dans le domaine de la **traduction automatique** : cette approche est utilisée depuis peu par Google pour améliorer Google Translate par exemple. Les « word embeddings » permettent en particulier d'identifier les correspondances entre termes techniques dans différentes langues, sans avoir recours à des linguistes ou à des traducteurs spécialisés sur un sujet pointu.
• L'analyse du **sentiment** : le word embedding permet de diminuer l'impact des tournures ambigües et permettent d'analyser plus précisément le caractère positif ou négatif des commentaires. Qui plus est, le word embedding permet également d'éviter de créer des systèmes de reconnaissances de patterns, généralement créés à la main. Le processus devient plus automatique.
• La **reconnaissance vocale** : le word embedding apporte également de nouvelles méthodes pour créer les bases d'information nécessaires à une bonne efficacité de la reconnaissance vocale .

Quel rapport entre Rankbrain, Word2vec et le "word embedding" ?

Voici deux exemples donnés par Google où Rankbrain est censé avoir permis d'améliorer la compréhension de la requête, et donc la qualité des résultats :

• « What is the label of a consumer at the highest level of a food chain » : la question appelle le concept de … prédateur.
• « Can you get 100% score on Super Mario without walkthrough » : ici le problème c'est la négation « without », qui était mal comprise par le moteur avant Rankbrain.

La communication de Google sur Rankbrain montre qu'il s'agit bien d'une implémentation du word embedding dans l'algorithme du moteur.

Cela signifie en fait que dans Rankbrain, l'intelligence artificielle (en fait les réseaux de neurones) est sans doute utilisée pour calculer les représentations de terme, et c'est tout. Mais c'est déjà beaucoup, vu les progrès que permet cette méthode !

Certains observateurs, dans le monde du SEO, ont pensé que Rankbrain était une nouvelle couche d'intelligence introduite dans l'algorithme de classement. La réalité a l'air plus prosaïque : il s'agit d'embarquer, dans une architecture de moteur de recherche somme toute classique, un nouveau « modèle de langue » (une nouvelle façon de modéliser le langage). Mais rien que cela permet d'améliorer la précision du moteur.

Quelles leçons en tirer pour le SEO ?

Pour le moment, l'implémentation du word embedding dans le moteur Google en est à ses débuts. Rankbrain a certes amélioré la qualité des résultats renvoyés sur certaines requêtes, mais cette implémentation des possibilités offertes par le word embedding reste limitée, et basique.

Rankbrain est juste une nouvelle preuve qu'il devient de moins en moins nécessaire de chercher à se positionner sur un mot clé précis pour être visible en tête des résultats. Rankbrain permettra de faire remonter des pages contenant des synonymes ou des contenus équivalents. C'est juste un outil de plus qui transforme Google en « moteur de réponses », où les questions appellent des réponses qui sont des concepts, et plus des pages web dont le contenu est similaire aux mots clés de la requête. Mais Rankbrain n'est qu'une étape de plus dans cette direction, entamée officiellement avec Hummingbird. Cette évolution est très, très progressive, mais c'est bien la direction vers laquelle les équipes de Google vont.

Et demain ?

Le word embedding est un nouveau modèle, qui est réellement employé et étudié depuis quelques années seulement. Il s'agit d'une avancée tout à fait notable dans un champ de recherche qui n'en avait plus connu depuis longtemps.

Les précédents modèles ont mis des années avant d'être exploités complètement. Nous sommes donc peut-être au début de la découverte de nouvelles applications du concept, qui peuvent à terme révolutionner tous les domaines d'application du traitement automatique des langues, et en particulier, des moteurs de recherche.

En attendant, on peut être à peu près sûr qu'étudier ce nouveau paradigme sera incontournable pour continuer à faire du SEO dans les années qui viennent, car il y a de fortes chances que les anciennes méthodes d'optimisation se basant sur le vieux modèle vectoriel embarqué dans les moteurs fonctionnent de moins en moins bien.

Les changements sur les moteurs s'accélèrent, il va donc falloir s'adapter, et comprendre le mieux possible ce nouveau modèle.

Philippe YONNET , *Directeur Général de l'agence Search-Foresight, groupe My Media (http://www.search-foresight.com)*